UNITED STATES PATENT APPLICATION TRANSMITTAL FORM

Box: NEW PATENT APPLICATION
ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Docket No.: **909.0001 USU**

"Express Mail" mailing label number: **EL 338 490 117 US**
Date of Deposit: Sept 22, 1999

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and addressed to the Asst. Commissioner for Patents, Washington, D.C. 20231.

Suzanne Bates
(Name of person mailing paper)

Suzanne Bates
(Signature of person)

Sir:

Transmitted herewith for filing is the patent application of

Inventor: **David M. Chess**

For: **METHOD AND APPARATUS FOR INCREASING VIRUS DETECTION SPEED USING A DATABASE**

Enclosed are:

_X_ Declaration and Power of Attorney and an Associate Power of Attorney;

_X_ **27** pages of Specification, Claims and Abstract;

_X_ **3** sheet of drawings;

_X_ An Assignment of the invention to: International Business Machines Corporation, Armonk NY U.S.A., with Assignment Recordation Form;

___ The certified copy of a priority application;

_X_ Information Disclosure Statement (and Form PTO-1449 with copies of cited documents);

___ Verified Statement of Small Entity Status (Non-Profit Institution);

___ Verified Statement of Small Entity Status (Small Business Concern).

___ Priority of Provisional Application No. _____, filed on _____ is herewith claimed under 35 U.S.C. §119(e).

The Filing Fee is calculated below.

CLAIMS AS FILED

(1) For	(2) Number Filed	(3) Number Extra	(4) Rate	(5) Basic Fee
				$760.00
Total Claims	**37** - 20 =	**17**	$ 18.00	**$306.00**
Independent Claims	**4** - 3 =	**1**	$ 78.00	**$ 78.00**
Assignment Recordal Fee:				**$ 40.00**
TOTAL FILING FEE				**$1184.00**

1/2 FILING FEE FOR SMALL ENTITY

_X_ A check in the amount of **$1184.00** in payment of the filing fee is enclosed.

___ Charge $ _____ to Deposit Account No. 01-0467.

Fee Deficiency: The Commissioner is hereby authorized to charge any additional fees under 37 C.F.R. 1.16 and 1.17 which may be required for this communication or during the entire pendency of this patent application, or credit any overpayment, to Deposit Account No. 01-0467.

Address all future communications to:

Harry F. Smith, Esq.
OHLANDT, GREELEY, RUGGIERO & PERLE, L.L.P.
One Landmark Square
Suite 903
Stamford, CT. 06901

Direct phone calls to Harry F. Smith at (203) 327-4500

9/22/99
Date of Signature

Harry F. Smith
Harry F. Smith, Esq.
Attorney for Applicant(s)
Ohlandt, Greeley, Ruggiero & Perle
Registration No. 32,493
(203) 327-4500

EXPRESS MAIL NO.: EL 338 490 117 US
International Business Machines Corporation Docket No.: YO999-078
Ohlandt, Greeley, Ruggiero & Perle, L.L.P. Docket No.: 909.0001 USU
Patent Application Papers of: David M. Chess

METHOD AND APPARATUS FOR INCREASING VIRUS DETECTION SPEED USING A DATABASE

FIELD OF THE INVENTION:

This invention is generally related to data processing systems and methods and, more particularly, to techniques for detecting the presence of undesirable software entities, such as computer software viruses, in data processing systems.

BACKGROUND OF THE INVENTION:

One technique to protect a data processing system against computer viruses and other undesirable software entities is to periodically scan the potentially infectable objects (e.g., applications, files, etc.) on the system for the presence of known viruses, or new viruses that are sufficiently similar to known viruses to be detected using available algorithms. However, this process can be time-consuming, especially as the size of data processing systems and the number of known viruses increases.

More particularly, existing anti-virus software makes use of a large variety of algorithms to detect the presence of computer viruses and other undesirable software entities (hereinafter simply referred to as "viruses".) As the size of a typical system increases, and the number and complexity of known viruses and the objects that they

infect increases, the time required to check a typical system for viruses also increases. Various techniques for speeding up these checks are known in the art. In general, most of these known techniques involve improved algorithms for deciding whether a given object contains a virus, independent of any information about the object other than its current contents.

It is known to employ techniques for increasing the speed of virus scanning by maintaining a database of information about the status of scanned objects, at the time the last scan was performed, and then using that database to determine which objects are new, or have changed in significant ways, since the last scan. Reference in this regard can be had to U.S. Patent No.: 5,473,769, "Method and Apparatus for Increasing the Speed of the Detecting of Computer Viruses", By Paul D. Cozza. Scanning only these objects can significantly reduce the time taken to perform the scan. However, this technique is not effective when one or more new viruses have been added to the set being scanned for. That is, since the new viruses were not scanned for the last time, the fact that an object has not changed since the last scan cannot be taken as indicating that the object need not be scanned this time. This is true since, if the object contains one of the newly-added viruses, the last scan would not have detected the new virus, but the current scan will. These known techniques, then, do not convey any advantage when the list of viruses being scanned for is updated between scans. As new computer viruses continue to appear more and more frequently, and network connectivity makes it feasible to update the virus list more and more often, the effectiveness of these known techniques can be expected to decline significantly.

A general reference to computer virus detection and removal

techniques can be found in a publication coauthored by the inventor, "Fighting Computer Viruses", Scientific American, November 1997, J.O. Kephart et al., pp. 88-93. Reference may also be had to the following commonly assigned U.S. Patents for teaching various computer virus detection, removal and notification techniques: U.S. Patent No.: 5,440,723, issued 8/8/95, entitled "Automatic Immune System for Computers and Computer Networks", by Arnold et al.; U.S. Patent No.: 5,452,442, issued 9/19/95, entitled "Methods and Apparatus for Evaluating and Extracting Signatures of Computer Viruses and Other Undesirable Software Entities", by Kephart; U.S. Patent No.: 5,485,575, issued 1/16/96, entitled "Automatic Analysis of a Computer Virus Structure and Means of Attachment to its Hosts", by Chess et al.; U.S. Patent No.: 5,572,590, issued 11/5/96, entitled "Discrimination of Malicious Changes to Digital Information Using Multiple Signatures", by Chess; and U.S. Patent No.: 5,613,002, issued 3/18/97, entitled "Generic Disinfection of Programs Infected with a Computer Virus", by Kephart et al. The disclosures of these commonly assigned U.S. Patents are incorporated by reference herein in their entireties, in so far as the disclosures do not conflict with the teachings of this invention.

OBJECTS AND ADVANTAGES OF THE INVENTION:

It is a first object and advantage of this invention to provide an improved technique to scan for the presence of viruses in a data processing system.

It is a second object and advantage of this invention to provide an improved technique to scan for the presence of virus-infected objects, wherein the improved technique requires less time than conventional techniques.

It is another object and advantage of this invention to

provide a virus detection technique that uses a database to store additional information, beyond what is necessary to determine simply whether or not a particular object, such as a file, has changed, and to then employ this additional information to achieve a more rapid virus scan, even when the list of viruses being scanned for has changed since a previous scan.

SUMMARY OF THE INVENTION

The foregoing and other problems are overcome and the objects of the invention are realized by methods and apparatus in accordance with embodiments of this invention.

This invention provides a method and system for increasing the speed of object virus-scanning by storing intermediate results of one or more stages of the virus-scanning process in a database, and then using, in at least some cases, the stored information to avoid recalculating the same results.

The stored information may be used to determine, to a high probability, whether or not a current object has changed since it was last scanned, as well as further information that can be used in at least some cases to increase the speed of the virus detection operation. For example, the use of the stored information enables a virus-detecting program to avoid processing steps that would otherwise have to be carried out, or by using alternative faster processing steps that utilize the stored information.

Also stored on a computer storage device is a list of descriptions of known viruses and possibly also classes of known viruses, including for each virus or virus class a sufficient amount of information to allow the virus-detecting program to determine, for a given object, whether or not the object is significantly likely to be infected

with that virus, or with a virus belonging to that class.

A virus detection method is disclosed for use in a computer system that contains at least one object that may potentially become infected with a computer virus. The method has steps of providing a database for storing information that is descriptive of a state of the at least one object as it existed at a point in the past and, for an object that is indicated as having a current state that is described by the stored information, a step of programmatically examining the object for a presence of a computer virus while assuming that the current state of the object is the same as the state of the object as it existed at the point in the past. This method, a computer program stored on a tangible medium that implements the method, and a system constructed and operated in accordance with the method, can significantly increase the speed of object virus-scanning.

BRIEF DESCRIPTION OF THE DRAWINGS

The above set forth and other features of the invention are made more apparent in the ensuing Detailed Description of the Invention when read in conjunction with the attached Drawings, wherein:

Fig. 1 is a simplified block diagram depicting an exemplary computer system on which a preferred embodiment of a virus-detecting program operates;

Fig. 2 is a block diagram that illustrates the logical connectivity between the virus-detecting program of Fig. 1, a storage unit for potentially infectable objects, and a database containing virus descriptions in accordance with an aspect of this invention; and

Figs. 3A and 3B are logic flow diagrams depicting the operation of the virus detection system and method in accordance with an embodiment of this invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 is a block diagram showing an exemplary data processing or computer system 100 on which a preferred embodiment of the present invention operates. The computer system 100 includes a computer platform 102 having a hardware unit 103, and a virus-detecting program 101 that implements the methods depicted in Figs. 3A and 3B. The virus-detecting program 101 operates on the computer platform 102 and hardware unit 103. The hardware unit 103 typically includes one or more central processing units (CPUs) 104, a random access memory (RAM) 105 and an input/output (I/O) interface 106. Microinstruction code 107, for example a reduced instruction set, may also be included on the platform 102. Various peripheral components may be connected to the computer platform 102. Typically provided peripheral components include a display 109, a data storage device (e.g. tape or disk) 110, and a printing device 111. A link 112 may also be included to connect the system 100 to one or more other similar computer systems shown simply as the block 113. The link 112 is used to transmit digital information between the computers 100 and 113. An operating system 114 coordinates the operation of the various components of the computer system 100, and is also responsible for managing the various objects or files, and for recording certain information regarding same, such as date and time last modified, file length, etc. An example of computer system 100 is the IBM IntelliStation™ (IntelliStation is a trademark of the IBM Corporation). It is expected that those skilled in the art will be familiar with many equivalent computer systems 100, and the teachings of this

invention are not to be construed to be limited in any way to the specific architecture depicted in Fig. 1.

Referring now as well to Fig. 2, in a presently preferred embodiment of this invention the virus-detecting program 101 has access to a collection of potentially-infectable objects 115, to a set of descriptions of known viruses and classes of viruses 116 and, in accordance with an aspect of this invention, to a database 117 containing information recorded during a previous execution of the virus-detecting program 101. The collection of potentially-infectable objects 115 can comprise, by way of example, documents, files and boot records that are stored on the disk 110, and which may also possibly be resident in the RAM 105.

Referring to Fig. 3A, at Block 310 the virus-detecting program 101 treats in turn each of a subset of the members of the collection of potentially-infectable objects 115. At Block 302, each potentially-infectable object 115 is examined for viruses. If an object 115 is determined to be probably infected at Block 303, then the information as to the infection is recorded at Block 304. After all objects 115 have been inspected, the information recorded at Block 304 can be presented to the user at Block 305, or is otherwise employed. For example, in an alternative embodiment of this invention the information stored at Block 305 as to the infection of objects 115 is sent to a central server, and/or is used by an automatic clean-up process with little or no user intervention, and/or is sent to an administrator across the network link 112. This information can also be presented immediately to the user for action, without waiting for all remaining ones of the objects 115 to be examined first. A virus-detecting program 101 that operates in accordance with the teachings of this invention may also include various virus-removal

and repair algorithms that are known in the art.

Fig. 3B illustrates the detailed operation of the virus-checking portion (Block 302) of the algorithm of Fig. 3A. Beginning with Block 310, for each object 115 being examined the virus-detecting program 101 first examines the object 115, and compares its current state to information recorded in the database 117 or elsewhere, such as in tables maintained by the operating system 114. If the object 115 is determined to have changed since the information was recorded in the database 117, the stored information for this object 115 is marked as invalid at Block 311. For example, the time and date of file creation or last modification may not agree with what is stored in the database 117, or the file length information may be different. If there is any indication that the object 115 may have been changed or altered since the last execution of the virus-detecting program 101, then the information in the database 117 is invalidated, such as by toggling a valid/invalid bit that is associated with the stored information in the database 117.

Next, at Block 312 the database information for the current object 115 (e.g., a document object) is examined to determine if it contains valid information concerning the number and location of macros (or other units of active content) in the object 115, where a macro is generally considered as a program or program-like object that is embedded within the object. If valid macro-related information is not found, control passes to Block 313 where the virus-detecting program 101 analyzes the object 115 to determine, for example, a number, location, size, name, extent, and/or other attributes of contained macros, if any, in the document object, and then stores that information in the database 117 at Block 314. The location can be indicated, for example, as a byte count from a

reference point in the object, such as a number of bytes from the beginning of a file. Next, at Block 315 the virus-detecting program 101 locates the macros contained in the object 115, if any, using information from the database 117, and tests the macro(s) for viruses using the virus descriptions 116 and a suitable virus-detection method.

Reference in this regard may be had to commonly assigned U.S. Patent Application S.N. 09/041,493, filed 3/12/98, entitled "Automated Sample Creation of Polymorphic and Non-Polymorphic Macro Viruses", by Jean-Michel Y. Boulay, August T. Petrillo and Morton G. Swimmer, the disclosure of which is incorporated by reference herein in its entirety.

Next, at Block 316 the database information 117 for the current object 115 is examined to determine if it contains valid information concerning the presence and location of any archived objects (for example, ZIPped or ARJed files) that are stored as components within the current object. If the database 117 does not contain valid information regarding archived objects, at Block 317 the virus-detecting program 101 analyzes the object 115 to determine the presence, number and location of any archived objects, and stores this information into the database at Block 318. Again, the location of archived objects may be indicated as a byte count from the beginning of the object 115. Next, at Block 319 the archived objects, if any, which are potentially infected are located using the information found in the database 117 and are examined for the presence of viruses using the virus descriptions 116 and any suitable method known in the art.

Further in this regard, for at least some objects 115 that contain in an archived or combined form one or more other objects, the method may store information as to whether any of the contained objects are of a type that are required to

be scanned for viruses. The virus-detecting program 101 is responsive to this stored information in the database 117 to not re-determine and re-scan the contained objects if the information in the database 117 indicates that none of them need to be scanned.

Also for those objects 115 that contain in an archived or combined form one or more other objects, the method may store information as to, for example, the location, extent, and/or encoding-method of contained objects which should be scanned for computer viruses. In this case the virus-detecting program 101 is responsive to the stored information for reducing the amount of processing time that is required to extract the contained objects in order to scan them.

Continuing now at Block 320, the database 117 entry for the current object 115 is examined to determine if it contains valid information concerning those features of the object 115 that may serve as input to a neural-network component of the virus-detecting program 101 (assuming that one is present). Such features can include, by example, the file length, the frequency of distribution of all or certain op-codes, and whether certain instructions or instruction types are found in macros. General reference with regard to a neural network-based virus detection scheme can be found in a publication entitled "Neural Networks for Computer Virus Recognition", by G. Tasauro et al., IEEE Expert Magazine, Vol. 11, no. 4, 8/96, pp. 5-6. If the feature information is not located in the database 117, control passes to Block 321 where the current object 115 is analyzed to determine which features are present, and the feature information is then added to the database 117 at Block 322. The neural network virus detector 101a is then run at Block 323, using the feature information that was previously added to, or just added to, the database 117, to

determine the possible presence of viruses in the object 115.

Further in this regard, the virus detecting program 101 includes or is coupled to the neural network (NN) unit component 101a that employs features from an object 115 to be examined, evaluates those features using weights and connections, and outputs an indication of whether the object 115 may contain a virus. In this case, and for at least one of the objects 115, the method stores at least a subset of object-related features that are relevant to the inputs of the neural network component 101a, and the neural network component 101a then uses the stored subset of features, rather than re-extracting these features from the object 115 itself, if the database 117 indicates that the object 115 has not changed since the features were previously extracted, and if the set of features used during the current scan by the virus-detecting program contains at least one of the features previously stored.

It should be appreciated that by making use of the information stored in the database 117, the potentially time-consuming steps of analyzing the objects for macros, archived components, and neural-network input features may be avoided, resulting in a significant increase in performance. For example, and referring again to Fig. 3B, it can be appreciated that if the database 117 is found to contain valid information for a particular object 115, then Blocks 312, 315, 316, 319, 320 and 323 can be executed in succession, thereby avoiding the time-consuming object processing involved in the blocks 313, 317 and 321.

In alternative embodiments of this invention, other types of stored information can be employed in the same way. For example, in one alternative embodiment the database 117 is used to record only whether a particular object 115

contains any macros, but not to record the location or other information regarding the macros. In this case the overall speed advantage is less, since if an object is specified as containing a macro then the location(s) must still be determined. However, an advantage that is gained is that the size of the database 117 can be reduced. In another embodiment, the database 117 is used to record only whether or not each object 115 contains any potentially-infectable archived components, but not to record their location or other information. Again, the speed advantage is reduced, but the size of the database 117 is also reduced.

In further embodiments of this invention, still other types of information stored within the database 117 can be employed in the manner described thus far. The additional test and object processing step(s) are depicted generally in Fig. 3B as the Block 325. Several examples of these tests and object processing are now provided. These examples are not intended to be read in a limiting sense upon the practice of this invention.

For example, in a further alternative embodiment the database is used to record whether or not each object can possibly be infected with a virus of any kind, according to certain criteria, and during execution of the virus-detecting program 101 any object 115 for which there is a valid record in the database 117 indicating that it cannot possibly be infected according to those criteria, and the criteria are the same as, or more stringent than, the criteria in effect for this particular run of the virus-detecting program 101, then the object 115 is not checked for viruses.

In a further embodiment of this invention the information stored in the database 117 is descriptive of the location

of an entry-point of the object 115, In this case the virus-detecting program 101 uses the stored information, rather than re-reading the object 115 to determine the entry-point, so long as the database 117 indicates that the object 115 has not changed since the entry-point information was last recorded.

In a further embodiment of this invention the information stored in the database 117 is descriptive of the location, size, and/or attributes of segments or other units of program-structure contained in the object 115. In this case the virus-detecting program 101 uses the stored information, rather than re-reading the object 115 to determine these aspects of its structure, so long as the database 117 indicates that the object 115 has not changed since the program structure information was last recorded.

It is also within the scope of the teachings of this invention to programmatically examine a current object by performing a program-emulation step that executes the current object in a virtual environment, and that collects or accumulates data resulting from the execution in the virtual environment. This data can include, by example, the contents of decrypted buffers and/or certain system calls issued by the object. In this case the virus-detecting program 101 includes or is coupled to program-emulation (PE) unit or component 101b, and stores at least some of the results data produced by the program-emulation unit, and then uses the stored information, and inhibits operation of the program-emulation unit 101b, if the database 117 indicates that the object 115 has not changed since the information was recorded.

Reference with regard to execution of an object or a portion of an object in a virtual environment may be had to commonly assigned U.S. Patent Application S.N. 09/160,117,

filed 9/24/98, entitled "Interpreter with Virtualized Interface", by David M. Chess, the disclosure of which is incorporated by reference herein in its entirety.

Based on the foregoing description it should be apparent that an aspect of this invention is a computer program that is embodied on a computer-readable medium, such as the disk 110, for providing a virus detection program subsystem. The computer program has a capability to create or at least maintain the database 117 for storing information that is descriptive of a state of at least one object 115 as it existed at a point in the past, and an object examination code segment 101 that is responsive to a determination that an object 115 has a current state that is described by the stored information in the database 117, for programmatically examining the object 115 for a presence of a computer virus, while using the stored information from the database 117. The computer readable medium further stores a list 116 comprised of information that is descriptive of at least one of known viruses and of known classes of viruses. The list 116 is used by the object examination code segment 101 when programmatically examining the object 115 for the presence of a computer virus. The computer readable medium may further store the neural network-based virus detection code segment 101a. In this case the database 117 stores further information that is descriptive of features of the object 115 that serve as inputs to the neural network-based virus detection code segment 101a. The computer readable medium may further store a program-emulation code segment 101b for executing objects 115 in a virtual environment, for collecting data resulting from the execution in the virtual environment, and for storing at least some of the results in the database 117. In this case the object examination unit code segment 101 is responsive to the stored results for using the stored results, and for inhibiting the operation of the

program emulation unit code segment 101b if said database indicates that the object 115 has not changed since the results were stored.

In a further aspect of these teachings the computer program is capable of executing a method for use in the computer system that includes the at least one potentially infectable object, where the method executed by the computer program has steps of, (a) maintaining a database comprised of the stored information descriptive of a state of the at least one object as it existed at a point in the past, and, for an object that the database indicates has a current state that is described by the stored information, programmatically examining the object for a presence of a computer virus while assuming that the current state of the object is the same as the state of the object as it existed at the point in the past.

It should be realized that the teachings of this invention are not intended to be limited to only the hardware and software architectures described above, nor to the storage in the database 117 of only the specific information that was described above. In general, the database 117 can be used to store any object-related information that can be used to facilitate the operation of the virus-detecting program or code segment 101. It should also be noted that certain steps of the method shown in Fig. 3 could be executed in other than the order shown (e.g., Blocks 312 and 315 could be interchanged with Blocks 316 and 319), other steps could be added, and other steps may be deleted. For example, if the neural network unit 101a is not present, then the execution of Blocks 320 through 323 is not required.

Thus, while the invention has been particularly shown and described with respect to preferred embodiments thereof, it

will be understood by those skilled in the art that changes in form and details may be made therein without departing from the scope and spirit of the invention.

CLAIMS

What is claimed is:

1. A virus detection method for use in a computer system comprising at least one object that may potentially become infected with a computer virus, comprising steps of:

providing a database comprised of stored information that is descriptive of a state of the at least one object as it existed at a point in the past; and

for an object that is indicated as having a current state that is described by the stored information, programmatically examining the object for a presence of a computer virus while assuming that the current state of the object is the same as the state of the object as it existed at the point in the past.

2. A method as in claim 1, wherein the stored information is descriptive at least in part of a number and location of macros within the object.

3. A method as in claim 1, wherein the stored information is descriptive at least in part of a number and location of archived objects within the object.

4. A method as in claim 1, wherein the stored information is descriptive at least in part of features of the object that serve as inputs to a neural network-based virus detection system, and wherein the step of programmatically examining the object comprises a step of operating the neural network-based virus detection system using the features as inputs.

5. A method as in claim 1, wherein the stored information is descriptive at least in part of whether at least one macro is present within the object.

6. A method as in claim 1, wherein the stored information is descriptive at least in part of whether at least one archived object is present within the object.

7. A method as in claim 1, wherein the stored information is descriptive at least in part of whether the object can possibly be infected with a virus according to predetermined criteria, and wherein the step of programmatically examining is executed only if the stored information indicates that the object may possibly be infected according to the predetermined criteria as compared to criteria that are currently in effect.

8. A method as in claim 1, wherein if it is indicated that a current state of the object is not described by the stored information, the step of programmatically examining comprises an initial step of processing the object to ascertain the current state of the object, and storing information in the database that is descriptive of the current state of the object.

9. A method as in claim 1, wherein the stored information comprises, for an object that contains in archived or combined form at least one other object, information descriptive of whether the at least one contained object is of a type that should be examined for computer viruses, and wherein the step of programmatically examining avoids re-determining and re-scanning the contained at least one object if the stored information indicates that the at least one contained object is not required to be scanned.

10. A method as in claim 1, wherein the stored information comprises, for an object that contains in archived or combined form at least one other object, information descriptive of at least one of a location, extent, or encoding-method of the at least one contained object, and wherein the step of programmatically examining is responsive to the stored information for reducing an amount of processing time required to extract the at least one contained object in order to examine the at least one contained object.

11. A method as in claim 1, wherein the stored information comprises information descriptive of a location of an entry-point of the object, and wherein the step of programmatically examining uses the stored information to determine the entry-point of the object, if the database indicates that the object has not changed since the entry-point information was stored.

12. A method as in claim 1, wherein the stored information comprises information descriptive of a structure of the object, and wherein the step of programmatically examining uses the stored information to determine the structure of the object, if the database indicates that the object has not changed since the structure information was stored.

13. A method as in claim 1, wherein the stored information comprises information descriptive of at least one of a number, size, name, extent, or other attribute of macros or other units of active content in the object, and wherein the step of programmatically examining uses the stored information to determine at least one of the number, size, name, extent, or other attribute of macros or other units of active content in the object, if the database indicates that the object has not changed since the

information was stored.

14. A method as in claim 1, wherein the step of programmatically examining includes a program-emulation step for executing the current object in a virtual environment, for collecting data resulting from the execution in the virtual environment, and for storing at least some of the results produced by the program-emulation step in the database, and wherein the step of programmatically examining uses the stored results rather than re-executing the program-emulation step, if the database indicates that the object has not changed since the results were stored.

15. A virus detection component for use in a computer system that stores at least one object that may potentially become infected with a computer virus, comprising:

a database comprised of stored information that is descriptive of a state of the at least one object as it existed at a point in the past; and

an object examination unit bidirectionally coupled to said database and responsive to a determination that an object has a current state that is described by the stored information, for programmatically examining the object for a presence of a computer virus while using the stored information from said database.

16. A virus detection component as in claim 15, wherein the stored information is descriptive at least in part of a number and location of macros within the object.

17. A virus detection component as in claim 15, wherein the stored information is descriptive at least in part of a number and location of archived objects within

the object.

18. A virus detection component as in claim 15, wherein the stored information is descriptive at least in part of features of the object that serve as inputs to a neural network-based virus detection system, and wherein said neural network-based virus detection system uses the features as inputs.

19. A virus detection component as in claim 15, wherein the stored information is descriptive at least in part of whether at least one macro is present within the object.

20. A virus detection component as in claim 15, wherein the stored information is descriptive at least in part of whether at least one archived object is present within the object.

21. A virus detection component as in claim 15, wherein the stored information is descriptive at least in part of whether the object can possibly be infected with a virus according to predetermined criteria, and wherein said object examination unit programmatically examines said object only if the stored information indicates that the object may possibly be infected according to the predetermined criteria as compared to criteria that are currently in effect.

22. A virus detection component as in claim 15, wherein if said determination indicates that a current state of the object is not described by the information stored in said database, said object examination unit first processes the object to ascertain the current state of the object, and stores information in said database that is descriptive of the current state of the object.

23. A virus detection component as in claim 15, wherein the stored information comprises, for an object that contains in archived or combined form at least one other object, information descriptive of whether the at least one contained object is of a type that should be examined for computer viruses, and wherein said object examination unit inhibits re-determining and re-scanning the contained at least one object if the stored information indicates that the at least one contained object is not required to be scanned.

24. A virus detection component as in claim 15, wherein the stored information comprises, for an object that contains in archived or combined form at least one other object, information descriptive of at least one of a location, extent, or encoding-method of the at least one contained object, and wherein said object examination unit is responsive to the stored information for reducing an amount of processing time required to extract the at least one contained object in order to examine the at least one contained object.

25. A virus detection component as in claim 15, wherein the stored information comprises information descriptive of a location of an entry-point of the object, and wherein said object examination unit is responsive to the stored information to determine the entry-point of the object, if the database indicates that the object has not changed since the entry-point information was stored.

26. A virus detection component as in claim 15, wherein the stored information comprises information descriptive of a structure of the object, and wherein said object examination unit is responsive to the stored information for determining the structure of the object, if

the database indicates that the object has not changed since the structure information was stored.

27. A virus detection component as in claim 15, wherein the stored information comprises information descriptive of at least one of a number, size, name, extent, or other attribute of macros or other units of active content in the object, and wherein said object examination unit is responsive to the stored information for determining at least one of the number, size, name, extent, or other attribute of macros or other units of active content in the object, if the database indicates that the object has not changed since the information was stored.

28. A virus detection component as in claim 15, and further comprising a program-emulation unit for executing the current object in a virtual environment, for collecting data resulting from the execution in the virtual environment, and for storing at least some of the results produced by the program-emulation unit in said database, and wherein said object examination unit is responsive to the stored results for using said stored results, and for inhibiting the operation of said program emulation unit, if the database indicates that the object has not changed since the results were stored.

29. A computer program embodied on a computer-readable medium for providing a virus detection program subsystem, comprising:

a code segment for at least maintaining a database that stores information that is descriptive of a state of at least one object as the object existed at a point in the past; and

an object examination code segment that is responsive to a determination that the object has a current state that is described by the stored information in said database, for programmatically examining the object for a presence of a computer virus while using the stored information from said database.

30. A computer program as in claim 29, wherein said computer readable medium further stores a list comprised of information that is descriptive of at least one of known viruses and of known classes of viruses, said list being used by said object examination code segment when programmatically examining the object for the presence of a computer virus.

31. A computer program as in claim 29, wherein said computer readable medium further stores a neural network-based virus detection code segment, wherein said database further stores information descriptive of features of the object that serve as inputs to said neural network-based virus detection code segment, and wherein said neural network-based virus detection code segment uses the features as inputs.

32. A computer program as in claim 29, wherein said computer readable medium further stores a program-emulation code segment for executing objects in a virtual environment, for collecting data resulting from the execution in the virtual environment, and for storing at least some of the results in said database, and wherein said object examination unit code segment is responsive to the stored results for using said stored results, and for inhibiting the operation of said program emulation unit code segment, if said database indicates that the object has not changed since the results were stored.

33. A computer program embodied on a computer-readable medium, the computer program being capable of executing a method for use in a computer system that comprises at least one object that may potentially become infected with a computer virus, the method executed by the computer program comprising steps of:

maintaining a database that is comprised of stored information that is descriptive of a state of the at least one object as it existed at a point in the past; and

for an object that the database indicates has a current state that is described by the stored information, programmatically examining the object for a presence of a computer virus while assuming that the current state of the object is the same as the state of the object as it existed at the point in the past.

34. A computer program as in claim 33, wherein the stored information is descriptive at least in part of a number and location of macros within the object.

35. A computer program as in claim 33, wherein the stored information is descriptive at least in part of a number and location of archived objects within the object.

36. A computer program as in claim 33, wherein the computer program implements or has access to a neural network-based virus detection system, wherein the stored information is descriptive at least in part of features of the object that serve as inputs to the neural network-based virus detection system, and wherein the step of programmatically examining the object comprises a step of operating the neural network-based virus detection system using the features as inputs.

37. A computer program as in claim 33, wherein for an object that the database indicates has a current state that is not described by the stored information, the step of programmatically examining comprises an initial step of operating the stored program to process the object to ascertain the current state of the object, and storing information in the database that is descriptive of the current state of the object.

METHOD AND APPARATUS FOR INCREASING VIRUS DETECTION SPEED USING A DATABASE

ABSTRACT OF THE DISCLOSURE

A virus detection method is disclosed for use in a computer system that contains at least one object that may potentially become infected with a computer virus. The method has steps of providing a database for storing information that is descriptive of a state of the at least one object as it existed at a point in the past and, for an object that is indicated as having a current state that is described by the stored information, a step of programmatically examining the object for a presence of a computer virus while assuming that the current state of the object is the same as the state of the object as it existed at the point in the past. The virus detection method, system and computer program uses the database to store additional information, beyond what is necessary to determine simply whether or not a particular object, such as a file, has changed, and to then employ this additional information to achieve a more rapid virus scan, even when the list of viruses being scanned for has changed since a previous scan.

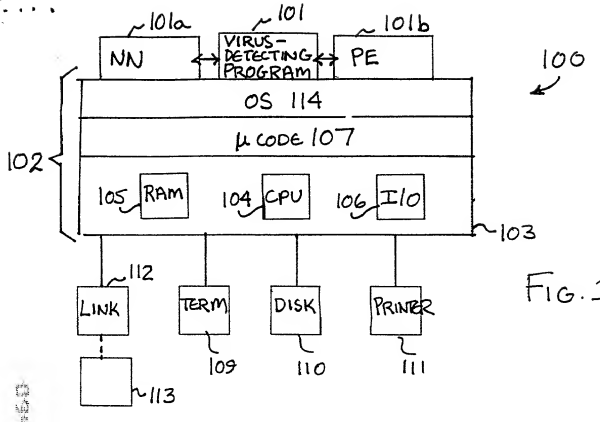
101a
101
101b
NN
VIRUS-DETECTING PROGRAM
PE
100
OS 114
μ CODE 107
102
105 RAM
104 CPU
106 I/O
103
112
LINK
TERM
DISK
PRINTER
113
109
110
111

FIG. 1

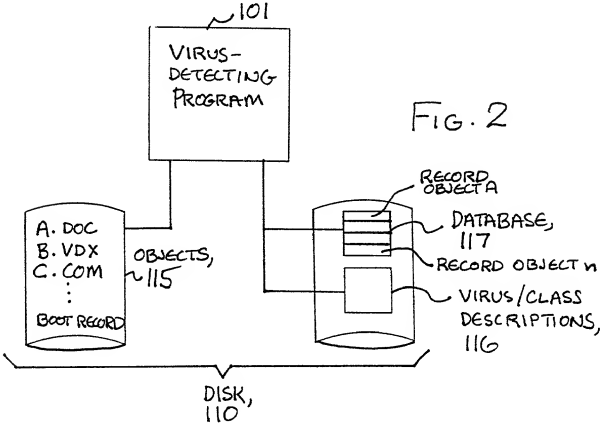
101
VIRUS-DETECTING PROGRAM
A. DOC
B. VDX
C. COM
BOOT RECORD
OBJECTS, 115
RECORD OBJECT A
DATABASE, 117
RECORD OBJECT n
VIRUS/CLASS DESCRIPTIONS, 116
DISK, 110

FIG. 2

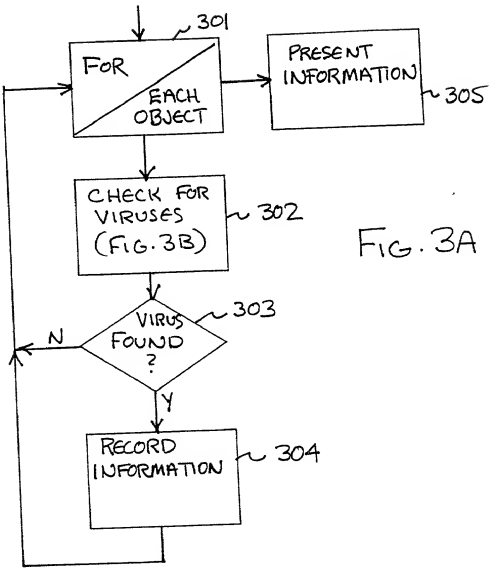
301
FOR
EACH
OBJECT
PRESENT
INFORMATION
305
CHECK FOR
VIRUSES
(FIG. 3B)
302
VIRUS
FOUND
?
303
N
Y
RECORD
INFORMATION
304

FIG. 3A

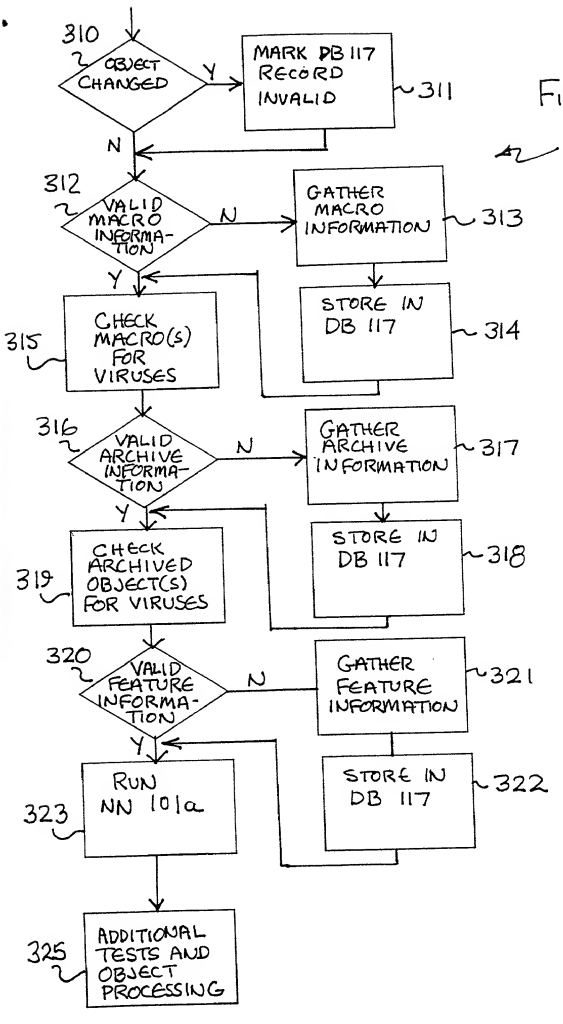

FIG. 3B

STEP 302 (FIG. 3A)

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

Docket No. **YO999-078**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND APPARATUS FOR INCREASING VIRUS DETECTION SPEED USING A DATABASE

the specification of which

(check one) __**X**__ is attached hereto.

_____ was filed on as Application Serial No.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, 119 of any foreign application(s) for patent or inventor's certificate(s) listed below and have also identified below any foreign application(s) for patent or inventor's certificate(s) having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
(Number)	(Country)	(Day/Mon/Year Filed)	___ Yes ___ No
(Number)	(Country)	(Day/Mon/Year Filed)	___ Yes ___ No
(Number)	(Country)	(Day/Mon/Year Filed)	___ Yes ___ No

I hereby claim the benefit under Title 35, United States Code, 119(e) and 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Prov. Appl. No.)	(Filing Date)	(Status) (patent, pend., abandon.)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

NAMES	REGISTRATION NUMBERS
Manny W. Schechter	Reg. No. 31,722
Terry J. Ilardi	Reg. No. 29,936
John E. Hoel	Reg. No. 26,279
Edward A. Pennington	Reg. No. 32,588
Christopher A. Hughes	Reg. No. 26,914
Joseph C. Redmond, Jr.	Reg. No. 18,753
Daniel P. Morris	Reg. No. 32,053
Douglas W. Cameron	Reg. No. 31,596
Louis P. Herzberg	Reg. No. 41,500
Kevin M. Jordan	Reg. No. 40,277
Stephen C. Kaufman	Reg. No. 29,551
Paul J. Otterstedt	Reg. No. 37,411
Louis J. Percello	Reg. No. 33,206
Jay P. Sbrollini	Reg. No. 36,266
David M. Shofi	Reg. No. 39,835
Robert M. Trepp	Reg. No. 25,933

SEND CORRESPONDENCE TO:

Harry F. Smith, Esq.
OHLANDT, GREELEY, RUGGIERO & PERLE, L.L.P.
One Landmark Square
Suite 903
Stamford, CT. 06901

DIRECT TELEPHONE CALLS TO:

Harry F. Smith, Esq.
Telephone:(203)327-4500
Facsimile:(203)327-6401

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

FULL NAME OF INVENTOR	LAST NAME	FIRST NAME	MIDDLE INITIAL
	CHESS	**DAVID**	**M.**
RESIDENCE & CITIZENSHIP	CITY	STATE OR COUNTRY	CITIZENSHIP
	MOHEGAN LAKE	**NEW YORK**	**U.S.A.**

POST OFFICE ADDRESS	P.O. ADDRESS	CITY & STATE	ZIP CODE
	1744 LAWRENCE ROAD	**MOHEGAN LAKE, NY**	**10547**

Inventor's Signature _____ Date 20 Sept 1999

Express Mail No.: EL 338 490 117 US

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
INVENTOR: David M. Chess
SERIAL NO.: 09/ ART UNIT:
FILED: Herewith EXAMINER:
TITLE: METHOD AND APPARATUS FOR INCREASING VIRUS DETECTION SPEED USING A DATABASE
ATTORNEY DOCKET NO.: YO999-078

Hon. Commissioner of Patents and Trademarks
Washington, D.C. 20231

APPOINTMENT OF ASSOCIATE ATTORNEY

Dear Sir:

The undersigned attorney, who has been appointed as an attorney in the Declaration and Power of Attorney for the above-identified patent application, hereby appoints:

Harry F. Smith, Esq.
Reg. No.: 32,493
OHLANDT, GREELEY, RUGGIERO & PERLE, L.L.P.
One Landmark Square
Suite 903
Stamford, CT. 06901

his associate attorney to prosecute said application and to transact all business in the United States Patent and Trademark Office connected therewith.

Please direct all official communications and telephone calls to:

Harry F. Smith, Esq.
OHLANDT, GREELEY, RUGGIERO & PERLE, L.L.P.
One Landmark Square
Suite 903
Stamford, CT. 06901

Telephone: (203)327-4500
Facsimile: (203)327-6401

Respectfully submitted,

9/21/99
Date

David M. Shofi (Reg. No.: 39,835)
IBM Corporation
IP Law Department
Yorktown Heights, New York 10598
(914) 945-3252